Amendment to the Specification:

Page 1, lines 4-26:

The present application claims the benefit under 35 USC § 119 of U.S. Patent application Serial Number 60/061,809, filed October 14, 1997, and U.S. Patent Application Serial Number 60/098,296, filed August 27, 1998, and claims the benefit under 35 USC § 120 of U.S. Patent Application Serial Number 09/067,544, filed April 27, 1998, U.S. Patent Application Serial Number 09/141,713, filed August 28, 1998, U.S. Patent Application Serial Number 09/384,792, filed August 27, 1999, U.S. Patent Application Serial Number 09/416,925, filed October 13, 1999, U.S. Patent Application Serial Number 09/439,603, filed November 12, 1999, U.S. Patent Application Serial Number 09/464,283, filed December 15, 1999, U.S. Patent Application Serial Number 09/514,425, filed February 28, 2000, U.S. Patent Application Serial Number 09/675,484, filed September 29, 2000, U.S. Patent Application Serial Number 09/675,700, filed September 29, 2000, U.S. Patent Application Serial Number 09/692,561, filed October 18, 2000, [[,]] U.S. Patent Application Serial Number ~~09/748,936561~~ 09/748,936, filed December 26, 2000, the U.S. Patent Application filed February 20, 2001, inventors Laurence B. Boucher et al., entitled "Obtaining a Destination Address so that a Network Interface Device can Write Network Data Without Headers Directly into Host Memory," the U.S. Patent Application filed March 7, 2001, inventors Peter K. Craft et al., entitled "Port Aggregation for Network Connections that are Offloaded to Network Interface Devices," the U.S. Patent Application filed March 9, 2001, inventors Laurence B. Boucher et al., entitled "Intelligent Network Interface System and Method for Protocol Processing," and the U.S. Patent Application filed March 9, 2001, inventors Stephen E. J. Blightman et al., entitled "Reducing Delays Associated with Inserting a Checksum into a Network Message," all of which are incorporated by reference herein.

Page 7, lines 10-11:

FIG. 15 is a diagram of a ~~Uniform~~ User Datagram Protocol (UDP) exchange that may transfer audio or video data between the client and server of FIG. 14.

Page 29, lines 7-20:

For the situation in which the file requested by client 602 was not present in a cache, but instead stored as file blocks on the server attached storage unit 634, the server 622 file system instructs a host SCSI driver to fetch the 100KB of data from server attached storage unit 634 into the server 600 file cache (assuming the file system does not wish to cache the data on INIC 622 file cache). The host SCSI driver then sends a SCSI request for the data over SCSI channel 638 to server attached storage unit 634. A controller on the server attached storage unit 634 responds to the commands by reading the requested blocks from its disk drive or drives and sending the blocks over SCSI channel 639 to the SCSI driver, which interacts with the cache manager under direction of the file system to store the blocks as file streams in the server 600 file cache. A file ~~systemredirector~~ system redirector then directs SMB to send a scatter-gather list of the file streams to INIC 622, which is used by the INIC 622 to read data packets from the server 600 file streams. The INIC 622 prepends the data packets with headers it created based on the server CCB, and sends the resulting frames onto network 604.

Page 30, line 14 – page 31, line 5:

The file system then orchestrates sending the file stream to server storage unit 634, network storage unit 640 or NAS storage unit 642. To send the file stream to server storage unit 634, the file system commands a SCSI driver in the server 600 to send the file stream as file blocks to the storage unit 634. To send the file stream to network storage unit 640, the file system directs the INIC to create ISCSI or similar headers based on the SAN CCB, and prepend those headers to packets read from the file stream according to the scatter-gather list, sending the resulting frames over network 644 to storage unit 640. To send the file stream to NAS storage unit 642, which may for example be useful in distributed file cache or proxy server implementations, the file ~~systemredirector~~ system redirector prepends appropriate NetBios/SMB headers and directs the INIC to create IP/TCP headers based on the NAS CCB, and prepend those headers to packets read from the file stream according to the scatter-gather list, sending the resulting frames over network 644 to storage unit 642.

FIG. 15 illustrates a system similar to that shown in FIG. 14, but FIG. 15 focuses on a ~~Uniform~~ User Datagram Protocol (UDP) implementation of the present invention, particularly in the context of audio and video communications that may involve Realtime Transport Protocol (RTP) and Realtime Transport Control Protocol (RTCP). Server 600 includes, in addition to the protocol layers shown in FIG. 14, a UDP layer 654, an AUDP layer 655, an RTP/RTCP layer 656 and an application layer 657. Similarly, client 602 includes, in addition to the protocol layers shown in FIG. 14, a UDP layer 660, an AUDP layer 661, an RTP/RTCP layer 662 and an application layer 663. Although RTP/RTCP layers 656 and 662 are shown in FIG. 15, other session and application layer protocols may be employed above UDP, such as session initiation protocol (SIP) or media gateway control protocol (MGCP).

Page 45, line 27 – page 47, line 31:

An array of sixteen comparators 1220 each receives the value stored in the corresponding block of the least-recently-used register 1200. Each comparator also receives a signal from processor ~~470~~ 780 along line 1235 so that the register block having a number matching that sent by processor ~~470~~ 780 outputs true to logic circuits 1230 while the other fifteen comparators output false. Logic circuits 1230 control a pair of select lines leading to each of the multiplexers, for selecting inputs to the multiplexers and therefore controlling shifting of the register block numbers. Thus select lines 1239 control MUX0, select lines 1244 control MUX7, select lines 1249 control MUX8, select lines 1254 control MUX9 and select lines 1259 control MUX15.

When a CCB is to be used, processor ~~470~~ 780 checks to see whether the CCB matches a CCB currently held in one of the sixteen cache units. If a match is found, the processor sends a signal along line 1235 with the block number corresponding to that cache unit, for example number 12. Comparators 1220 compare the signal from that line 1235 with the block numbers and comparator C8 provides a true output for the block R8 that matches the signal, while all the other comparators output false. Logic circuits 1230, under control from the processor ~~470~~ 780, use select lines 1259 to choose the input from line 1235 for MUX15, storing the number 12 in the MRU block R15. Logic circuits 1230 also send signals along the pairs of select lines for MUX8 and higher multiplexers, aside from

MUX15, to shift their output one block to the left, by selecting as inputs to each multiplexer MUX8 and higher the value that had been stored in register blocks one block to the right (R9-R15). The outputs of multiplexers that are to the left of MUX8 are selected to be constant.

If processor ~~470~~ 780 does not find a match for the CCB among the sixteen cache units, on the other hand, the processor reads from LRU block R0 along line 1266 to identify the cache corresponding to the LRU block, and writes the data stored in that cache to DRAM. The number that was stored in R0, in this case number 3, is chosen by select lines 1259 as input 1215 to MUX15 for storage in MRU block R15. The other fifteen multiplexers output to their respective register blocks the numbers that had been stored each register block immediately to the right.

For the situation in which the processor wishes to remove a CCB from the cache after use, the LRU block R0 rather than the MRU block R15 is selected for placement of the number corresponding to the cache unit holding that CCB. The number corresponding to the CCB to be placed in the LRU block R0 for removal from SRAM (for example number 1, held in block R9) is sent by processor ~~470~~ 780 along line 1235, which is matched by comparator C9. The processor instructs logic circuits 1230 to input the number 1 to R0, by selecting with lines 1239 input 1235 to MUX0. Select lines 1254 to MUX9 choose as input the number held in register block R8, so that the number from R8 is stored in R9. The numbers held by the other register blocks between R0 and R9 are similarly shifted to the right, whereas the numbers in register blocks to the right of R9 are left constant. This frees scarce cache memory from maintaining closed CCBs for many cycles while their identifying numbers move through register blocks from the MRU to the LRU blocks.

FIG. 24 illustrates additional details of INIC 22, focusing in this description on a single network connection. INIC 22 includes PHY chip 712, ASIC chip 700 and DRAM 755. PHY chip 712 couples INIC card 22 to network line ~~2105~~ 2102via a network connector 2101. INIC 22 is coupled to the CPU of the host (for example, CPU 30 of host 20 of Figure 1) via card edge connector 2107 and PCI bus 757. ASIC chip 700 includes a Media Access Control (MAC) unit 722, a sequencers block 732, SRAM control 744, SRAM 748, DRAM control 742, a queue manager 803, a processor 780, and a PCI bus

interface unit 756. Sequencers block 732 includes a transmit sequencer 2104, a receive sequencer 2105, and configuration registers 2106. A MAC destination address is stored in configuration register 2106. Part of the program code executed by processor 780 is contained in ROM (not shown) and part is located in a writeable control store SRAM (not shown). The program may be downloaded into the writeable control store SRAM at initialization from the host 20.

FIG. 25 is a more detailed diagram of the receive sequencer 2105 of FIG. 24. Receive sequencer 2105 includes a data synchronization buffer 2200, a packet synchronization sequencer 2201, a data assembly register 2202, a protocol analyzer 2203, a packet processing sequencer 2204, a queue manager interface 2205, and a Direct Memory Access (DMA) control block 2206. The packet synchronization sequencer 2201 and data synchronization buffer 2200 utilize a network-synchronized clock of MAC 722, whereas the remainder of the receive sequencer 2105 utilizes a fixed-frequency clock. Dashed line 2221 indicates the clock domain boundary.

Operation of receive sequencer 2105 of Figures 24 and 25 is now described in connection with the receipt onto INIC 22 of a TCP/IP packet from network line ~~702~~ 2102. At initialization time, processor 780 partitions DRAM 755 into buffers. Receive sequencer 2105 uses the buffers in DRAM 755 to store incoming network packet data as well as